OPIC
OFFICE DE LA PROPRIÉTÉ INTELLECTUELLE DU CANADA

CIPO
CANADIAN INTELLECTUAL PROPERTY OFFICE

(12) (19) (CA) **Demande-Application**

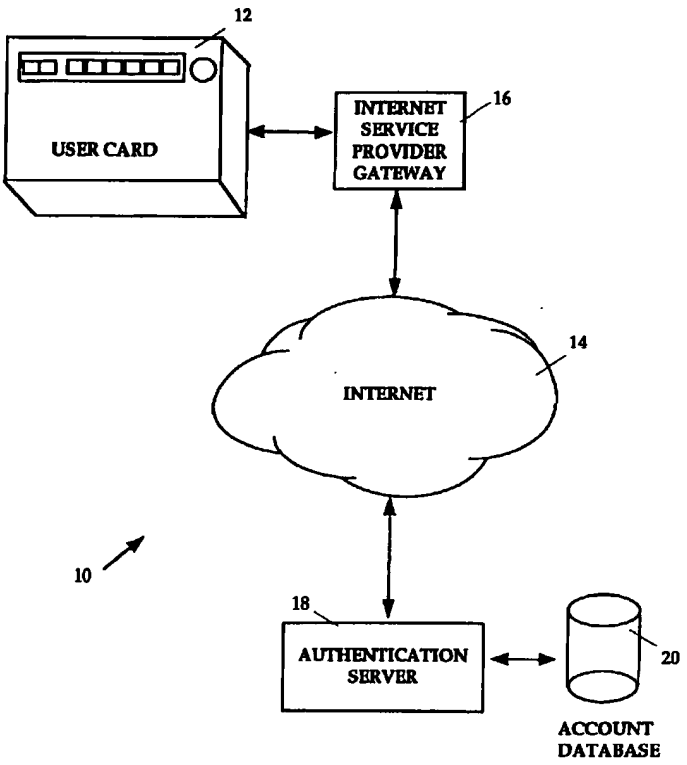
(21) (A1) **2,291,430**
(22) 1999/12/01
(43) 2000/07/28

(72) LU, TAO, CA
(71) LU, TAO, CA

(51) Int.Cl.[7] H04L 9/32, H04L 12/22
(30) 1999/01/28 (60/117,506) US
(30) 1999/02/15 (2,267,672) CA
(30) 1999/06/16 (09/336,483) US

(54) **SYSTEME DE SECURITE POUR LES TRANSACTIONS SUR INTERNET**

(54) **INTERNET TRANSACTION SECURITY SYSTEM**



(57) A method and apparatus for verifying that the bearer of a user card is authorized to use the card. A dynamic personal identification number (PIN) is used to provide improved security. The PIN comprises an event identifier and a pseudo-random number sequence identifier. On each transaction, the user card generates a new PIN by generating a distinct pseudo-random number based on a private seed and the previous random number stored by the user card and incrementing the value of the event identifier. The PIN is then transmitted to an authentication server along with a pre-established user account name. The authentication server then retrieves the private seed, and previous event and pseudo-random identifiers from a secure account database associated with the account name. The authentication server ensures that the stored event identifier corresponds to the event identifier provided by the user by incrementing the event identifier if necessary and by generating a successive pseudo-random identifier each time the event identifier is incremented. Once the event identifiers correspond, the latest pseudo-random identifier is compared with the pseudo-random identifier transmitted by the user within the PIN. If authentication is successful, the authentication server will then complete the financial transaction associated with the user's request.

ABSTRACT OF THE DISCLOSURE

A method and apparatus for verifying that the bearer of a user card is authorized to use the card. A dynamic personal identification number (PIN) is used to provide improved security. The PIN comprises an event identifier and a pseudo-random number sequence identifier. On each transaction, the user card generates a new PIN by generating a distinct pseudo-random number based on a private seed and the previous random number stored by the user card and incrementing the value of the event identifier. The PIN is then transmitted to an authentication server along with a pre-established user account name. The authentication server then retrieves the private seed, and previous event and pseudo-random identifiers from a secure account database associated with the account name. The authentication server ensures that the stored event identifier corresponds to the event identifier provided by the user by incrementing the event identifier if necessary and by generating a successive pseudo-random identifier each time the event identifier is incremented. Once the event identifiers correspond, the latest pseudo-random identifier is compared with the pseudo-random identifer transmitted by the user within the PIN. If authentication is successful, the authentication server will then complete the financial transaction associated with the user's request.

Title: INTERNET TRANSACTION SECURITY SYSTEM

FIELD OF THE INVENTION

This invention relates generally to systems for providing transactional security and more particularly to systems for providing improved security using a personal identification number (PIN).

BACKGROUND OF THE INVENTION

Personal identification systems which are based on the use of one or more of a card, badge and a memorized PIN number to ensure proper identification of an individual and to provide transaction security are well known. However, with the significant to provide increase in the amount of commerce which is conducted over the internet, traditional methods of securing transactions do not provide sufficient security to prevent unscrupulous merchants or other third parties from intercepting customer credit card data using electronic eavesdropping.

Specifically, little attention has been focussed on the problem of protecting the transaction facilitation information once the card has been authenticated, and in particular to the problem of misuse of the information by the merchant. Protection in this area has traditionally relied on the card owner's knowledge of the legitimacy of the merchant, which is reasonable when the card owner is at the point-of-sale. Protection is much less likely when the card owner is not at the point-of-sale, however, and the transaction is being carried out over the internet.

Generally, an internet "merchant" is often nothing more than an electronic address, and it is impossible for anyone to ensure that whoever is receiving the payment information is legitimate. Thus, such remote electronic transactions carry significant risks for both the customer and the credit provider. The customer is faced with the problem of misuse

of his or her account information, either by someone who has intercepted the information, or by a dishonest or compromised merchant, while the credit issuer is faced with the problem of verifying that a request for payment from a merchant is in response to a legitimate order.

A number of new security methods which have been developed for internet transaction use include CyberWallet, eCash, netCash and PayMe Transfer Protocols. Although these methods provide transaction security protection, they suffer from counterfeiting problems and a general reluctance on the part of government to accept new forms of financial currency. They also suffer from a number of drawbacks associated with their inherently complex nature which complicates installation within merchant and customer's hardware and which are not easily scalable.

For example, U.S. Patent No. 5,168,520 to Weiss describes a method and apparatus for providing improved security for a PIN number by having the user device and verification computer simultaneously mix a time dependent non-predictable code with a secret PIN according to a predetermined algorithm to produce combined coded values and then to compare these combined coded values to determine authentication status. While this reference attempts to conceal a user's PIN number by performing a type of secret encoding by mixing it within a time varying non-predictable code, the user's secret PIN is still transmitted over electronic communication media. Further, complex and expensive hardware must be utilized in order to provide the necessary synchronization of the user device and the verification computer. Also, since a new PIN is generated periodically in time, the system does not operate optimally over the internet where time delays (e.g. traffic jams) are commonplace. If the lifetime of each PIN is increased, the risk that the PIN will be intercepted and revised is also increased. If the system does

not allow reuse of a PIN, then the user will encounter further processing delays. Finally, since the PIN has to be generated continuously during the course of a day, the battery of the user device will run down at an accelerated rate.

Accordingly, there is a need for a simple, relatively inexpensive, easy to use transactional security system which does not transfer any sensitive data over the internet and which does not require the installation of complicated software or hardware by either the customer or the merchant.

SUMMARY OF THE PRESENT INVENTION

It is therefore an object of the present invention to provide an authentication system for confirming the identify of a user, said authentication system comprising:

(a) means for generating a unique pseudo-random identifier and for storing the unique pseudo-random identifier;

(b) means for incrementing an event identifier with each generation of the unique pseudo-random identifier;

(c) means for combining said pseudo-random identifier and said event identifier to produce a PIN; and

(d) means for transmitting to an authentication computer and account name and said PIN.

In a second aspect, the present invention provides a method for authenticating the identity of a user comprising the steps of:

(a) generating a pseudo-random identifier;

(b) generating an event identifier and incrementing

the value of the event identifier after generation of said pseudo-random identifier;

(c) combining the pseudo-random identifier with the event identifier to form a PIN; and

(d) communicating an account name for uniquely identifying a user and the PIN to an authentication computer for performing an authentication operation.

BRIEF DESCRIPTION OF THE DRAWING FIGURES

For a better understanding of the present invention and to show more clearly how it may be carried into effect, reference will now be made, by way of example, to the accompanying drawings, which show a preferred embodiment of the present invention and in which:

Fig. 1 is a schematic drawing of the basic components of an embodiment of the present invention;

Fig. 2A is a drawing showing a top view of the card of Fig. 1;

Fig. 2B is a schematic drawing of the circuitry contained within the card of Fig. 1;

Fig. 2C is a diagrammatic view of a card reader for reading the card of Fig. 1;

Fig. 2D is a diagrammatic view of the data table structure of account database of Fig. 1;

Fig. 3A is a flow chart diagram illustrating one embodiment of the MAIN SERVER routine used by the authentication server of Fig. 1 to effect authentication of the user;

Fig. 3B is a flow chart diagram illustrating one embodiment of the AUTHENTICATION routine used by the authentication server of Fig. 1 to effect authentication of the user;

Fig. 4A is a flow chart diagram illustrating one embodiment of the MAIN USER routine used by the user card of Fig. 1 to generate a PIN;

Fig. 4B is a flow chart diagram illustrating one embodiment of the GENERATE PIN routine used by the user card of Fig. 1 to generate a PIN;

Fig. 5 is a diagrammatic view of a typical internet financial transaction configuration;

Fig. 6 is a flow chart diagram illustrating one embodiment of the process used to complete a secured transaction within the internet financial configuration of Fig. 5;

Fig. 7 is a diagrammatic view of a typical cybermall shopping financial configuration;

Fig. 8 is a flow chart diagram illustrating one embodiment of the process used to complete a secured transaction within the internet financial configuration of Fig. 7;

Fig. 9 is a diagrammatic view of a typical software authentication configuration; and

Fig. 10 is a flow chart diagram illustrating one embodiment of the process used to complete a secured transaction within the software authentication configuration of Fig. 9.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Reference is first made to Fig. 1, which shows an authentication system 10 made in accordance with a preferred embodiment of the invention. A user card 12 is used in association with a conventional electronic processing and transmitting device (e.g. computer, enhanced wrist watch, electronic address book, palm pilot) (not shown), the device being coupled to the internet 14, and in particular the world wide web (WWW) via an internet service provider gateway 16. User card

12 may be employed by a user when searching the WWW in a conventional manner and provides communication data to an authentication server 18, which itself is connected to an account database 20.

Currently, if a user desires to purchase a product or service from a remotely located vendor, the user typically provides a credit card number to the vendor over the internet 14, thus potentially allowing a third party to access the card number, even if an encryption technique is employed. In accordance with the present invention, this problem is avoided because sensitive user data is never transmitted over the internet 14 and each transaction has a unique authorization number associated with it, as will be described.

According to the present invention, user card 12 is capable of generating and displaying a dynamic eight digit Personal Identification Number (PIN) number each time it is actuated by the user. The eight digit PIN comprises a scrambled combination of a two digit user generated variable USER_EVENT_ID and a six digit pseudo random user generated variable USER_RANDOM_ID. The variable USER_RANDOM_ID is generated using conventionally known random number generation algorithms, which generally use one or more privately kept SEED(S) and one or more previously generated random number(s). Each time user card 12 is triggered to generate a new PIN, the variable USER_EVENT_ID is incremented by one and a new variable USER_RANDOM_ID is generated. It should be understood that variables USER_EVENT_ID and USER_RANDOM_ID could be of any suitable digit length or radix as will become apparent. The combination of variables USER_EVENT_ID and USER_RANDOM_ID is then scrambled using a conventionally known encryption mask MASK to generate a PIN.

In order to initiate authentication with authentication server

18, the user will trigger user card 12 to generate and display a new PIN and then provide, or allow the vendor to provide, authentication server 18 with the user's Account Name (ACCTNAME) and the newly generated PIN. The user may provide the ACCTNAME and the generated PIN using a keypad or other data input device or verbally over the phone. It should be understood that in order for a user to obtain a user card 12, the user must provide personal information such as credit card number(s), date of birth, mother's maiden name, etc. by secure off-line methods to the credit administrator of the authentication server 18. The credit administrator would then grant the user a specific account name ACCTNAME and send the physical user card 12 to the user by secure means. Further, it should be understood that each new or unused user card 12 contains a specific user's "footprint" including the unique SEED and the first random number PREV_USER_RANDOM_ID. It should be understood that multiple SEEDS and/or multiple PREV_USER_RANDOM_IDs could be used to generate a new PIN.

Authentication server 18 maintains a variety of user-specific information in account database 20 so that it may authenticate a user's PIN for the purposes of authorizing a transaction over the internet 14. Specifically, the SEED utilized by the pseudo random number generator for each individual card is kept privately by authentication server 18 so that only the authentication server 18 will be able to reproduce the PIN generated by user card 12. For security purposes, the seed would not be transmitted across the internet 14 and would only be located within user card 12 and physically sent to user.

Accordingly, for each transaction authentication server 18 will look up the user's ACCTNAME in account database 20 and retrieve the associated SEED, the variable containing the previous user event ID (PREV_EVENT_ID) and the variable containing the previous user

random ID (PREV_RANDOM_ID) stored by authentication server 18 for that particular user. Authentication server 18 will then increment the variable PREV_EVENT_ID and generate a pseudo-random value to be stored on the authentication server in the temporary variable AS_RANDOM_ID based on a stored SEED value (identical to the one maintained by user card 12) and the variable PREV_RANDOM_ID. The value of PREV_EVENT_ID is stored on the authentication server in the temporary variable AS_EVENT_ID. If the new AS_EVENT_ID does not match the USER_EVENT_ID provided by the user, authentication server 18 will increment the variable AS_EVENT_ID and generate a corresponding pseudo-random value for the variable AS_RANDOM_ID using SEED and the latest generated variable AS_RANDOM_ID. This process is repeated until AS_EVENT_ID is equal to USER_EVENT_ID. Accordingly, variables USER_EVENT_ID and AS_EVENT_ID are used by user card 12 and authentication server 18 for synchronization purposes (i.e. in case the user has inadvertently over-triggered user card 12). Even in the rare case that user card 12 falls out of synchronization with authentication server 18, the user can re-synchronize user card 12 by calling the authentication company who administers authentication server 18 and providing them with the PIN shown on the display of the user card 12.

Once the USER_EVENT_ID and AS_EVENT_ID have been synchronized the variables AS_RANDOM_ID and USER_RANDOM_ID are compared. If AS_RANDOM_ID is equal to USER_RANDOM_ID then authentication server 18 will confirm the transaction and update the appropriate records of account database 20 to complete the transaction. Otherwise, authentication will be deemed to have failed.

Fig. 2A depicts user card 12 in a preferred embodiment as a smart card. As previously mentioned, the functionality of user card 12

could be implemented completely in software and installed within a user's computer or other electronic communication device such as a palm pilot and the like. User card 12 is typical in some respects to a conventional credit card. It has a thin sheet material body with an approximate length and width of a standard credit card or smaller for user convenience. A display 22 is provided for displaying an eight digit PIN number which is formed by scrambling the EVENT_ID and RANDOM_ID variables. A push button switch 28 is provided which is activated each time the card is used. When switch 28 is depressed an electric signal is produced which activates user card 12 to generate a new PIN for display on display 22.

Unlike conventional credit cards, user card 12 has the circuit shown in Fig. 2B encased within the card. Imbedded within user card 12 is a microprocessor 30, a power source 32, and a counter 34.

Microprocessor 30 may be any commercially available programmable device suitable for implementation within user card 12 (e.g. having relatively small dimensions). Storage of program instructions and other static data is provided by read only memory (ROM) 36, while storage of dynamic data is provided by a random access memory (RAM) 38. Both memory units 36 and 38 are accessed by microprocessor 30. Power source 32 provides operational power to microprocessor 30 as well as display 22.

Counter 34 is connected to microprocessor 30 and counts each time the card is used, that is, each time switch 28 is depressed by the user and accordingly, increments the variable USER_EVENT_ID. It should be understood that user card 12 could be password protected to ensure that the PIN number will only be generated and displayed when a user having the correct password is operating user card 12. This would ensure that user card 12 cannot be operated by someone who knows the card-holder and the associated ACCTNAME.

Microprocessor 30 is programmed to generate a pseudo-random number value for variable USER_RANDOM_ID for a corresponding USER_EVENT_ID based on a SEED and PREV_USER_RANDOM_ID, as discussed. The following C++-type pseudo code illustrates the broad concept of random number generation:

```
typedef PIN_type unsigned long;
PIN_type random(PIN_type SEED, PIN_type PREV_RANDOM_ID)
{
//Generates a pseudo random number according to previous number, there could be multiple
//ways to generate random numbers. This invention has no bias against any random number
//generator algorithm as long as the new random number is generated based on the previous
//random number generated and one or several seed number(s) distinct from an individual
//smart card.

RANDOM_ID = (SEED * PREV_RANDOM_ID) mod 0xffffff   //use last six hex-digits
return RANDOM_ID;
}
```

A conventional clock or timer 40 is activated for a predetermined time period, for example, 30 to 60 seconds each time user card 12 is used. When this time period has elapsed, timer 40 is automatically turned off and the PIN is cleared from display 22.

As shown in Fig. 2C, in an alternative embodiment, user card 12 can be adapted for use with a conventional card reader 42 to assist the user in entering the PIN during the course of a transaction. Card reader 42 may be installed in association with a computer or a POS terminal, as is conventionally known. Switch 28 of user card 12 includes a photodiode sensor 44 and user card 12 also consists of an LED 46 to transmit a light-encoded signal representing the PIN. Correspondingly, card reader 42 would use photodiodes 48, 50 and an LED 52. If the user device 12 is not in the card reader 42 then photosensor 44 will be off. When user card 12 is inserted into card reader 42, photosensor 44 will detect light from the LED

52 of card reader 42.

Photosensor 44 can be used to trigger a gate of an internal transistor Q_1 (not shown) "on" so that the serial PIN signal generated by microprocessor 30 of user card 12 can be used to modulate the operation of LED 46 to emit light pulses representing the PIN. The insertion of user card 12 into card reader 42 also breaks the line of sight between LED 52 and photodiode 50, triggering card reader 42 to enter a "read" state. Once card reader 42 is in read state, since LED 46 is in close proximity with photodiode 48, the light emitting pulses representing the PIN will be transmitted between LED 46 of user card 12 and photodiode 48 of card reader 42. Card reader 42 will then decode this light pulse train signal into digital form and provide it to the user's computer, POS or other such device, as is conventionally known.

As shown in Fig. 2D, database 20 is preferably implemented as a relational database comprising a user table 60 which contains essential user specific "footprint" information such as ACCTNAME, PREV_AS_EVENT, PREV_AS_RANDOM_NUMBER, SEED and unique encryption MASK, as well as other user account information for credit processing purposes, such as credit card account numbers, address, phone, birthday, credit rating and other user information typically held by a credit facility.

The user table 60 contains the user master record which will include all individual user related data fields. An example layout of such a user master record is as follows:

User Record Fields

ACCTNAME	5 characters (alphanumeric)
SEED	10 characters (numeric)
MASK	8 characters (numeric)

PREV_USER_EVENT_ID	2 characters (numeric)
PREV_USER_RANDOM_ID	6 characters (numeric)
Address I.D.	30 characters (alphanumeric)
City I.D.	3 characters (alphanumeric)
Province/State I.D.	3 characters (alphanumeric)
Country I.D.	3 characters (alphanumeric)
Credit Card I.D.	12 characters (numeric)

Depending on how authentication server 18 is configured (i.e. to operate as a third party credit authorization entity or as a part of a central multi-vendor facility, commonly known as a cyber-mall. A cyber-mall is an electronic version of a physical shopping mall. Multiple vendors exist at a single location. In the cyber-mall case, the user would access a single website to purchase a wide variety of articles from a multitude of vendors. Database 20 may contain a suitable merchant table 62 which contains merchant specific such as contact information and address. In such a case, merchant table 62 will contain the merchant master record for each vendor. The format of these records will be common to all merchants which subscribe to the authorization system. An example layout of such an merchant master record would be as follows:

Merchant Record Fields

Merchant Identification	10 characters (numeric)
Business Name	20 characters (alphanumeric)
Contact Name	20 characters (alphanumeric)
Business Address	32 characters (alphanumeric)
E-mail for Business	10 characters (alphanumeric)
Bank	10 characters (alphanumeric)
SEED	10 characters (numeric)
MASK	8 characters (numeric)

As described above, if authentication server 18 is configured to operate as a central multi-vendor facility, database 20 should then also contain a suitable product/service table 64 which contains product/service

specific information such as product description, availability and pricing. Each product/service record would be related to a particular merchant. An example layout of such an product/service master record would be as follows:

Product/Service Record Fields

Product/Service I.D.	10 characters (numeric)
Description of Product	20 characters (alpha-numeric)
Merchant I.D.	10 characters (numeric)
Price	5 characters (decimal)
Number in Stock	10 characters (numeric)

Finally, if authentication server 18 is configured to operate as a central multi-vendor facility, database 20 should also contain an accounting table 66 which is to be constantly up-graded as new payments are processed. This information would be used for making a complete accounting to the various merchants which would be hosted by authentication server 18. An example layout of such an product/service master record would be as follows:

Accounting Record Fields

Transaction Number	10 characters (numeric)
Date of Transaction	8 characters (numeric)
Amount of Transaction	5 characters (decimal)
Product/Service I.D.	20 characters (alpha-numeric)
Credit Card Number	12 characters (numeric)
Authorization Number	12 characters (numeric)

It should be understood that the fields of user table, merchant table, product/service table and accounting table, could be of any suitable length or data type, as conventionally understood.

Figs. 3A and 3B show flow chart diagrams which illustrate a

preferred embodiment of the MAIN SERVER and AUTHENTICATION routines performed by authentication server 18 to authenticate the identity of a user by comparing the PIN provided by user card 12 with server generated values. Each block of Figs. 3A and 3B identifies an operation to be performed by authentication server 18 to provide the functionality contemplated by the present invention.

With respect to the MAIN SERVER routine, authentication server 18 rests in an idle state at step 100. Upon receiving an electronic data message from a user card 12 at step 102, authentication server 18 parses a received electronic data message from user card 12 to obtain the user transmitted PIN and ACCTNAME at step 104. At this point, the user's IP_ADDRESS can be identified by authentication server 18 (not shown). At step 106, the AUTHENTICATION routine is called with the parameters PIN, and ACCTNAME. The AUTHENTICATION routine will perform authentication of the user's PIN based on the historical information stored within account database 20 that is retrieved using ACCTNAME and return either TRUE or FALSE. If TRUE is returned, the authentication server 18 continues at step 110 with the necessary communication necessary to complete the transaction. However, if FALSE is returned, authentication server 18 will return back to an idle state at step 100 and await further incoming data messages.

As shown in Fig. 3B, when AUTHENTICATION routine is called at step 106 (with the parameters PIN and the user ACCTNAME), authentication server 18 performs a lookup in the account database 20 and at step 113 retrieves a number of data records in the user data table associated with ACCTNAME, namely PREV_AS_EVENT_ID and PREV_AS_RANDOM_ID as well as a simple encryption MASK and the user's pseudo-random number generator SEED.

At step 114, authentication server 18 performs a simple decryption

of the user provided PIN by bitwise exclusive OR'ing the PIN with MASK. The variable MASK is unique to each user and is used to provide a basic level of encryption security. Further, at step 116, authentication server 18 obtains the USER_EVENT_ID and USER_RANDOM_ID components from PIN by bitwise AND'ing the PIN with an "1" bit string of appropriate length and by performing a left register shift to obtain the first two digits of PIN (the EVENT_ID).

At step 118, authentication server 18 sets the variable AS_EVENT_ID to the value of PREV_AS_EVENT_ID which was retrieved from account database 20 of authentication server 18 and generates the variable AS_RANDOM_ID based on PREV_AS_RANDOM_ID and SEED. By initially updating the variables AS_EVENT_ID and AS_RANDOM_ID, attempts to reuse the PIN by intercepting third parties will not result in authorization by authentication server 18. At step 120, authentication server 18 enters into a synchronization loop if the variable AS_EVENT_ID is not equal to the variable USER_EVENT_ID obtained from the user transmitted PIN.

If the synchronization loop is entered into, at step 122 the variable PREV_AS_RANDOM_ID is set to the value of the variable AS_RANDOM_ID. At step 124, a new pseudo-random variable AS_RANDOM_ID is generated using PREV_AS_RANDOM_ID and SEED. At step 126, the variable AS_EVENT_ID is incremented by 1 within a certain range (e.g. using the mod function to create a repeatable series of values representable within the size chosen for EVENT_ID). As long as the variable AS_EVENT_ID is not equal to USER_EVENT_ID, a new pseudo-random value will be generated for variable AS_RANDOM_ID based on the newly updated PREV_RANDOM_ID and SEED and the AS_EVENT_ID incremented until it is found to be equal USER_EVENT_ID.

Once AS_EVENT_ID is equal to USER_EVENT_ID, authentication is determined at step 128 based on whether the generated variable AS_RANDOM_ID is equal to USER_RANDOM_ID. If it is not, authorization fails at step 130 and FALSE is returned by the AUTHENTICATION routine. The transaction will be discarded (i.e. the values of PREV_AS_EVENT_ID and PREV_AS_RANDOM_ID will not be altered in account database 20). If AS_RANDOM_ID is equal to USER_RANDOM_ID, authorization succeeds at step 128 then at step 132 PREV_AS_EVENT_ID is equated to AS_EVENT_ID and PREV_AS_RANDOM_ID is equated to AS_RANDOM_ID and both values are updated in account database 20 for use in the next transaction. Finally, at step 134, TRUE is returned by the AUTHENTICATION routine.

It should be noted that the for loop represented by steps 120 to 126 ensures that if the variable AS_EVENT_ID does not match the user generated variable USER_EVENT_ID, authentication server 18 will increment the variable AS_EVENT_ID and generate a corresponding AS_RANDOM_ID based on the last generated RANDOM_ID. This is done in case the user has inadvertently over-triggered user card 12 and user card 12 is not synchronized with authentication server 18.

Accordingly, the variable EVENT_ID is used to synchronize (or "catch up") authentication server 18 with user card 12. It should however, be noted that if the users triggers the PIN so that USER_EVENT_ID runs a full complete cycle through the preset series of values for EVENT_ID, this program will perform the match of AS_RANDOM_ID and USER_RANDOM_ID without performing any "catch up" (as AS_EVENT_ID and USER_EVENT_ID will be apparently equal). This can be rectified practically, by setting the length of the cycle to be reasonably long or by using a recovery program to synchronize authentication server 18 with user card 12, once this occurs. In the rare case that the user over-

triggers user card 12 through the preset series of values for a complete cycle, synchronization will not be possible. The user will be required to contact the administrator of authentication server 18 and provide the PIN currently displayed by user card 12 to the administrator. The administrator will be able to decrypt the USER_EVENT_ID and the USER_RANDOM_ID and manually update account database 20 with these new values so that user card 12 can be re-synchronized with authentication server 18.

The following C++-type pseudo code illustrates an implementation of the MAIN SERVER and AUTHENTICATION routines, discussed above in relation to Figs. 3A and 3B.

```
main()
{
receive data  message
parse received message to get PIN, ACCTNAME and other information
if  (PIN_authentication(ACCTNAME,PIN))
{
 Authentication successful, do further process;
} else {
  Authentication failed
}
return;
};

Bool PIN_authentication(ACCTNAME,PIN)
{
//This code will check if the PIN is correct
retrieve PREV_AS_EVENT_ID, PREV_AS_RANDOM_ID, MASK and SEED from
authentication server's database according to customer's ACCTNAME;
PIN=PIN ^ MASK; //Exclusive or PIN with MASK to perform simple decryption
USER_RANDOM_ID=PIN & 0xffffff; //Bitwise AND to detach RANDOM_ID from PIN
USER_EVENT_ID=(PIN>>24);  //left shift PIN by 24 to get EVENT_ID from PIN
for (PIN_type count=PREV_AS_EVENT_ID;count!=USER_EVENT_ID; count=(count+1)
mod 0xff)
{
        randomnumber = random (SEED, PREV_AS_RANDOM_ID);
```

```
};
AS_RANDOM_ID=random (SEED, randomnumber);
if (AS_RANDOM_ID==USER_RANDOM_ID)
{
        //Authentication succeeds
        PREV_AS_RANDOM_ID=USER_RANDOM_ID;
        PREV_AS_EVENT_ID=USER_EVENT_ID;
        save PREV_AS_RANDOM_ID, PREV_AS_EVENT_ID in account database;
        return TRUE;
} else {
        //Authentication fails
        return FALSE;
};
};
```

Figs. 4A and 4B show a flow chart diagram that illustrates a preferred embodiment of the MAIN USER and GENERATE PIN routines used by user card 12 to generate a new PIN which will be used to initiate authorization of a transaction by authentication server 18. Each block of Figs. 4A and 4B identifies an operation to be performed by user card 12 to provide the functionality contemplated by the present invention. It should be noted that the operations performed by user card 12 may be implemented programically by software residing in microprocessor 30 or by direct electrical connections through customized integrated circuits or by a combination of both.

With respect to the MAIN USER routine, user card 12 rests in an idle state at step 200 until switch 28 is activated (e.g. either physically depressed or triggered by an LED signal from card reader 42). When switch 28 of user card 12 is activated, user card 12 is programmed to receive a PASSWORD from the user at step 202 (e.g. entered using an alphanumeric keypad (not shown) on user card 12) to safeguard card use, as is conventionally known. If PASSWORD is not found to correspond to a USER_PASSWORD stored in ROM memory 36 of user card 12 at step 204,

then user card 12 will return to idle state. If PASSWORD does correspond, then the GENERATE PIN routine will be called at step 206. It should be understood that the USER_PASSWORD is initially stored in ROM memory 36 at the time that user card 12 is manufactured. When PIN is returned by the GENERATE PIN routine, the PIN is send for display for a predetermined period of time, after which user card 12 will return to the idle state. As previously described, PIN is displayed so that user may enter PIN manually into an input device such as a keyboard or verbally by telephone.

Once the GENERATE PIN routine has been called static values MASK and SEED are retrieved from ROM memory 36 and the variables PREV_EVENT_ID and PREV_USER_RANDOM_ID are retrieved from RAM memory 38 at step 208. At step 210, the value of PREV_USER_EVENT_ID is increased by one and saved as USER_EVENT_ID. At step 212, USER_RANDOM_ID is generated using PREV_USER_RANDOM_ID and SEED within a conventionally known pseudo-random number generation algorithm, as previously described. At step 214, the user PIN is created from the values of USER_EVENT_ID and USER_RANDOM_ID by combining them bitwise.

At step 216, PIN is encrypted by exclusively OR'ing the PIN with an encryption MASK. At step 218, PREV_USER_EVENT_ID is updated to equal the value of USER_EVENT_ID and PREV_USER_RANDOM_ID is updated to equal the value of USER_RANDOM_ID. At step 220, the GENERATE PIN routine returns PIN, which is displayed on display 22, as previously described for a predetermined period of time (e.g. 30 or 60 seconds), after which user card 12 returns to the MAIN USER routine at step 200.

The following C++-type pseudo code illustrates the implementation of the MAIN USER and GENERATE PIN routines, discussed above in

relation to Figs. 4A and 4B.

```
main()
{
//main program of the smart card. For increased security, the card can be password
protected.

get a password from the user
if the password is not correct
 exit from program
else
 PIN=PIN_generation();
end;
return 0;

PIN type PIN_ generation()
{
//This code will generate a new PIN according to PREV_USER_RANDOM_ID and
//PREV_EVENT_ID associated with it. MASK is used for simple encryption of the PIN.
//The MASK, SEED and the first PREV_USER_RANDOM_ID are the footprints of an
//individual smart card. They should be kept privately and confidentially by smart card
//and authentication server.

Retrieve PREV_USER_RANDOM_ID, PREV_USER_EVENT_ID, MASK and SEED from
memory
PIN_typeUSER_EVENT_ID=(PREV_USER_EVENT_ID+1) mod 0xff; //increase count by 1
PIN _type USER_RANDOM_ID=random(SEED, PREV_USER_RANDOM_ID);
//generate new random number
USER_EVENT_ID<<24;  //left shift count by 24 to combine with USER_RANDOM_ID
PIN_type PIN=USER_EVENT_ID| USER_RANDOM_ID;  //combine them by bitwise or
PIN=PIN ^ MASK;  //simple encryption by exclusive or PIN with mask
save USER_EVENT_ID, USER_RANDOM_ID as PREV_USER_EVENT_ID,
PREV_USER_RANDOM_ID
return PIN;
}
```

Figs. 5 and 6 illustrate an embodiment of the present invention wherein a typical internet financial transaction (e.g. on-line shopping) is contemplated. The financial transaction involves a buyer 300, a seller 302, authentication server 18 and account database 20. Typically, buyer 300 will "visit" the virtual store of seller 302 using a typical web browser (e.g.

Netscape's Navigator or Microsoft's Internet Explorer) which can support the Secure Socket Layer (SSL) security protocol or other secure communication means.

Seller 302 posts product information to the browser of buyer 300. If buyer 300 wants to proceed with the purchase of one or more products or services featured, buyer 300 then responds by sending a confirmation message including his or her ACCTNAME to seller 302 at step 304. At step 306, seller 302 generates SELLER PIN. At step 308, seller 302 forwards BUYER ACCTNAME, SELLER ACCTNAME, SELLER PIN TRANSACTION AMOUNT and redirect the buyer's browser to the IP address of authentication server 18 to validate the transaction. The receipt of this information will cause authentication server 18 to initiate its MAIN SERVER routine.

At step 309, authentication server 18 executes its AUTHENTICATE routine using SELLER ACCTNAME and SELLER PIN. If authentication fails, at step 310, authentication server 18 sends a notice to seller 302 advising that seller authorization has failed. If authorization succeeds, then at step 312, authentication server 18 posts to buyer's web browser and requests the buyer's PIN along with final confirmation of the transaction.

At step 314, buyer 300 activates his or her user card 12 to generate a BUYER PIN. At step 316, buyer 300 sends the BUYER PIN to authentication server 18. At step 318, authentication server 18 executes AUTHENTICATE routine. If authentication fails, at step 320 authentication server 18 sends a notice to buyer 300 and seller 302 advising that authentication has failed. If authentication succeeds and TRANSACTION AMOUNT does not exceed the credit allowance of buyer 300, authentication server 18 at step 322 sends confirmation to seller 302 that authentication has been successful and confirms receipt of this confirmation by seller 302.

At step 324, authentication server 18 updates account database 20 to reflect the incremented EVENT_IDs and the newly generated RAMDOM_IDs created for both the buyer and seller account. Simultaneously, at step 326, seller 302 posts confirmation of completed transaction to buyer 302.

Figs. 7 and 8 illustrate another embodiment of the present invention wherein a typical cyber-shopping mall is contemplated. The financial transaction involves a number of buyers 400, sellers 402 and a cybermall 404. Authentication server 18 and account database 20 are located within cybermall 404. Typically, a buyer 400 will "visit" cybermall 404 (i.e. the authentication server 18 itself) where products and services offered by the sellers 402 will be displayed for sale.

It should be understood that once a seller has been approved for inclusion in the cyber-mall, the account database 20 will include specific data tables on various products, ordering, and stock information associated with sellers 402. Further, the authentication server 18 will be associated with an appropriate webpage display which will feature the products and services of the subscribing sellers 402. By locating a substantial amount of seller information on the authentication server 18, authentication server 18 may complete most of the transaction process on the sellers' behalf.

At step 406, authentication server 18 posts product information to buyer 400. At step 408, buyer 400 is instructed to provide a PIN and triggers user card 12 to execute the GENERATE PIN routine. Once generated, at step 410, the PIN is sent along with the buyer's ACCTNAME to authentication server 18 which initiates the MAIN SERVER routine. At step 412, authentication server 18 executes the AUTHENTICATION routine. If authentication fails then at step 414, authentication server 18 advises BUYER of failed authentication. If authentication succeeds then at step 416, authentication server 18 completes the transaction with buyer

400. Finally, at step 418, authentication server 18 sends a confirmatory order to seller 402 (includes Authorization Number) and confirms receipt of the confirmation by seller 402.

Figs. 9 and 10 illustrate another embodiment of the present invention wherein software installation authentication by a software customer 500 through manufacturer server 501 is contemplated. Currently, protection against software piracy is poor, as anyone with a copy of a CD key can easily install pirated software with little difficulty. In the present embodiment, customer 500 is assumed to own an authentication user card 12 and each software CD is distinguished by a serial code. The software in the CD is encrypted by conventional cryptography methods and the key can be stored in the CD or obtained from a manufacturer server 501.

The customer will initiate installation of software by accessing and running the CD setup program resident on the CD. At step 502, the installation program will query customer 500 to provide ACCTNAME and PIN. At step 504, customer 500 will generate and enter PIN using user card 12, as had been described. At step 506, the setup program sends ACCTNAME and PIN to manufacturer's server 501. At step 508, manufacturer server 501 forwards ACCTNAME and PIN to authentication server 18.

At step 510, authentication server 18 executes AUTHENTICATE routine and if authentication fails, at step 512, CD is considered to be pirated and setup program will terminate. If authentication succeeds, then at step 514 manufacturer's server 501 will receive notification from authentication server 18. At step 516, manufacturer server 501 sends a private key to the setup program so that the setup program may access the necessary information from manufacturer's server 501 and registers the CD according to its serial number. Finally, at step 518, setup program will

then decrypt the CD and install the software. It should be understood that the functionality of authentication server 18 could be incorporated into manufacturer's server 501.

It should be understood that it would also be possible to achieve authentication on a stand-alone computer by providing the user with a software CD to be protected and an accompanying user card 12. The installation software on the CD would have the SEED of the user card 12 built-in. When the user begins to install software from the software CD, the installation program will prompt the user for a PIN and the user would trigger user card 12 to generate a PIN based on the SEED and initial variables USER_RANDOM_ID and USER_EVENT_ID. When the user keys in the generated PIN into the installation program, the installation program will be able to authenticate that the user is installing a legally procured copy of the CD software by generating the same PIN using the SEED and its stored initial variables AS_RANDOM_ID and AS_EVENT_ID. If authentication is successful, the installation will complete installation and otherwise, it will terminate the installation

Another way to accomplish authentication of a software CD would be where each CD has its own distinct SEED. At the time of installation, the software installation program would provide the user with an unpredictable random number RANDOM. The user would then provide authentication server 18 with his or her ACCTNAME, a PIN generated by user card 12, and the number RANDOM generated by the installation program. Authentication server 18 would then check ACCTNAME and PIN to see whether the user is the legitimate owner of the CD. If so, then authentication server 18 will generate a new PIN by retrieving the SEED associated with the CD from the account database 20 and combining it with the number RANDOM using the relation embodied in the following C++-type pseudo code: PIN = (SEED* RANDOM) mod oxff. Authentication server 18

will then send the new PIN to the user. The user can then key in the PIN generated by authentication server 18 into the installation program. The installation program will use its built-in SEED and the number RANDOM which it initially provided to user to check whether the PIN is correct. If the PIN is correct then the installation program will continue installation. Otherwise, it will terminate installation.

Another way to accomplish authentication of a software CD would be where each CD has a unique SEED, maintained by the installation software and a vendor's authentication server 18. In such a system, the user would purchase a CD software product from a large vendor (e.g. Microsoft Corporation). The user would be required to provide a time/date dependent code CDKEY to the CD in order for the installation software to complete installation of the software. In order to obtain a valid code CDKEY, the user would be required to provide a user ACCTNAME and a PIN generated by the user's user card 12 as well as the CD serial number to authentication server 18 maintained by the vendor, preferably by way of a interactive webpage.

If authentication server 19 confirms the identity of the user based on the provided ACCTNAME and PIN, then authentication server 18 would generate the required code CDKEY using the generated SEED and the particular time/date associated with the user's request and provide the code CDKEY to the user through the webpage. Authentication server 18 could generate the code CDKEY using a relation embodied in the following C++-type pseudo code: CDKEY = (SEED* DATE) mod oxff. The user would then provide the code CDKEY to the installation software, which in turn would confirm whether the code CDKEY is correct or not. Since the installation software records the time/date of the user's request on the vendor's webpage, the installation software is able to generate the same CDKEY based on it's built-in SEED and on the time/date of the user

request and thus, is able to confirm that the user should be allowed to continue with software installation.

It should be understood that it would be possible to provide authentication without using the variables RANDOM_ID or EVENT_ID. Specifically, user card 12 would include a keypad and the user would be provided with a random number RANDOM by authentication server 18. The user would then input the number RANDOM provided by authentication server 18 into the user card 12. User card 12 would in turn generate a new PIN based on the relation embodied in the following C++-type pseudo code: PIN = (SEED* RANDOM) mod oxff. When user card 12 generates PIN, the PIN can then be authenticated by authorization server 18. Authentication server 18 would retrieve the SEED associated with user card 12 in account database 20 and combine it with the number RANDOM that it initially provided to the user, using the same relation used by user card 12.

Finally, it should be understood that instead of having authentication server 18 act as an intermediary agent, it would be possible to use an inexpensive card packaged with the CD, which would use a random number RANDOM provided by the installation program and a built-in SEED to generate a new PIN for installation authentication by the installation program. Since the installation program will know the SEED assigned to the card and the number RANDOM which it has provided, the installation program will be able to confirm that the PIN provided by the card and keyed in by the user is correct and that the CD is an authentic version.

Although the preferred embodiment of the present invention utilizes a pseudo-random number generator to create the random ID's, a random ID need not be restricted to a number. It may be any form of alphanumeric data that is readable by a user and replicable by a repeatable

generator. Similarly, the event ID need not be numeric in the traditional decimal sense, it merely needs to be able to store a value in any format that can be incremented over a range of values and decoded to represent the number of times a user random ID has been generated.

It should be appreciated that further application of the present invention may be made in the context of other e-commerce transactions, namely a customer could transfer funds to another registered customer's account on the Internet using user card 12. It would also be possible to generate a cybercheck to an unregistered Internet user by issuing a cybercheck payable to another person authenticated by a PIN. The printed cheque could be deposited by the receiver at a bank at which point the check could be cleared between the bank and the authentication server 18.

Further, the user card 12 of the present invention may also serve as an improved replacement of a conventional credit card and direct payment method. Typically, in a point-of-sale (POS) context, a user's credit card number or bank card number is read into a merchant's card reader and the user then keys in his or her (typically static) PIN for authentication. The merchant may intercept the card number and/or PIN for fraudulent purposes. Use of the present invention will avoid this problem as a merchant will not be able to fraudulently use the user provided information in view of the dynamic nature of the PIN.

Finally, the dynamic PIN arrangement can be used in remote access control by using user card 12 to trigger a new access PIN for each attempted account login. As is conventionally known, there is a substantial risk when users remote-login into their corporate network using a dial-up modem. Accordingly, strong authentication is required to ensure that improper access by a third party to a confidential host does not occur. By requiring the user to generate a new PIN each time the user attempts login according to the present invention, various unscrupulous eavesdropping

techniques (eg. viruses which the emulate a login window to steal password) can be averted.

Since user account information is stored in the private database of authentication server 18, no sensitive information will propagate over the Internet. Accordingly, the present invention eliminates the possibility that information such as date of birth, credit card numbers, and the like will be intercepted by a third party or obtained by a seller or merchant with which the user is transacting with. Further, the simplicity of the software required for installation by a user (i.e. either a buyer or a seller) allows for easy application of the invention.

As will be apparent to those skilled in the art, various modifications and adaptations of the method and system described above are possible without departing from the present invention, the scope of which is defined in the appended claims.

THE EMBODIMENTS OF THE INVENTION IN WHICH AN EXCLUSIVE PROPERTY OR PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

1. An authentication system for confirming the identify of a user, said authentication system comprising:

(a) means for generating a unique pseudo-random identifier and for storing the unique pseudo-random identifier;

(b) means for incrementing an event identifier with each generation of the unique pseudo-random identifier;

(c) means for combining said pseudo-random identifier and said event identifier to produce a PIN; and

(d) means for transmitting to an authentication computer and account name and said PIN.

2. The authentication system of claim 1, wherein the authentication computer comprises:

(a) means for utilizing the account name to retrieve a previous pseudo-random identifier and a previous event identifier, and a pseudo-random generating seed;

(b) means for generating an authenticating pseudo-random identifier, based upon said previous pseudo-random identifier and previous said event identifier; and

(c) means for comparing said authenticating pseudo-random identifier and said pseudo-random identifier provided by user to perform an authorization operation.

3. The authentication system of claim 1, wherein the means for combining said pseudo-random identifier and said event identifier to produce a PIN consists of a bitwise combination of said pseudo-random identifier and said event identifier.

4. The authentication system of claim 3, wherein the means for combining said pseudo-random identifier and said event identifier further consists of a mask encryption of said bitwise combination of said pseudo-random identifier and said event identifier.

5. The authentication system of claim 2, wherein the means for generating an authenticating pseudo-random identifier that corresponds to the event identifier provided by the user, comprises:

(a) means for utilizing the previous event identifier and generating a series of incremented authenticating event identifiers starting with the previous event identifier until the last of said series of said authenticating event identifers is equal to the event identifier provided by the user; and

(b) means for using the pseudo-random number generating seed to generate a series of successive authenticating pseudo-random identifiers, each successive authenticating pseudo-random identifier being based on

a previously generated authenticating pseudo-random identifier and said seed, and corresponding to an unique authenticating event identifier, such that the last of said series of successive authenticating pseudo-random identifiers corresponds to the last of said series of authenticating event identifers.

6. The authentication system of claim 2, wherein the means for using said authenticating pseudo-random identifier and said pseudo-random identifier provided by user to perform an authorization operation comprise means for bitwise comparing said last of said series of successive authenticating pseudo-random identifiers and said pseudo-random identifier.

7. A method for authenticating the identity of a user comprising the steps of:

(a) generating a pseudo-random identifier;

(b) generating an event identifier and incrementing the value of the event identifier after generation of said pseudo-random identifier;

(c) combining the pseudo-random identifier with the event identifier to form a PIN; and

(d) communicating an account name for uniquely identifying a user and the PIN to an authentication computer for performing an authentication operation.

8. The method of claim 7, wherein the authentication computer utilizes the account name to retrieve a previous pseudo-random identifier

and a previous event identifier, both previously received from the user, obtaining said pseudo-random identifier and said event identifier from said PIN and generating an authenticating pseudo-random identifier that corresponds to the event identifier provided by the user, to perform an authentication operation.

9. The method of claim 8, wherein the authentication computer utilizes the previous event identifier and generates a series of successively incremented authenticating event identifiers starting with the previous event identifier until the last of said series of authenticating event identifers is equal to the event identifier provided by the user.

10. The method of claim 9, wherein the authentication computer utilizes the user account name to retrieve a pseudo-random number generating seed and generates a series of successive authenticating pseudo-random identifiers, each successive authenticating pseudo-random identifier based on a previously generated authenticating pseudo-random identifier and each successive authenticating pseudo-random identifier corresponding to an unique authenticating event identifier, such that the last of said series of successive authenticating pseudo-random identifiers corresponds to the last of said authenticating event identifers.

11. The method of claim 10, wherein the authentication operation further comprises comparing said pseudo-random identifier and the last of said series of authenticating pseudo-random identifiers.

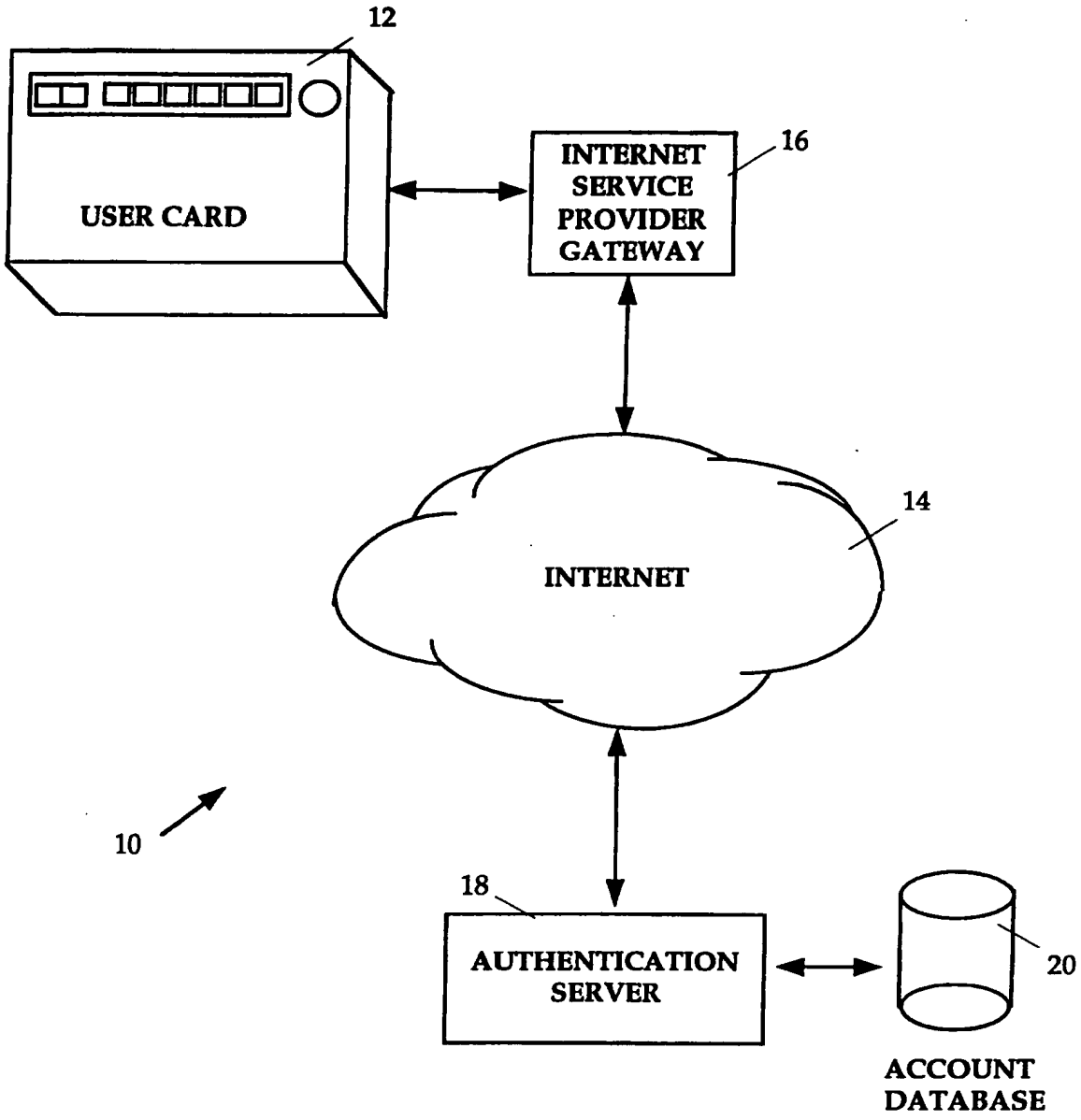

12
USER CARD
INTERNET SERVICE PROVIDER GATEWAY
16
14
INTERNET
10
18
AUTHENTICATION SERVER
20
ACCOUNT DATABASE

FIG. 1

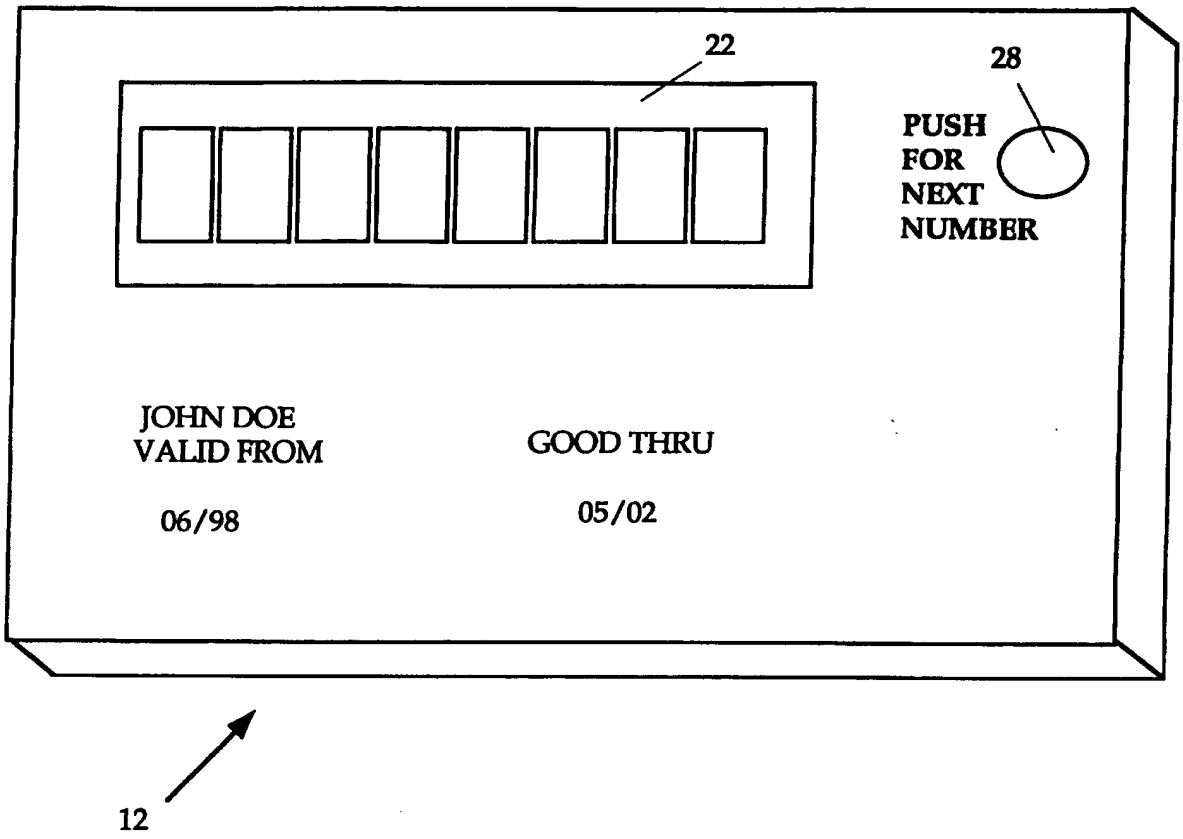

22
28
PUSH
FOR
NEXT
NUMBER
JOHN DOE
VALID FROM
06/98
GOOD THRU
05/02
12

FIG. 2A

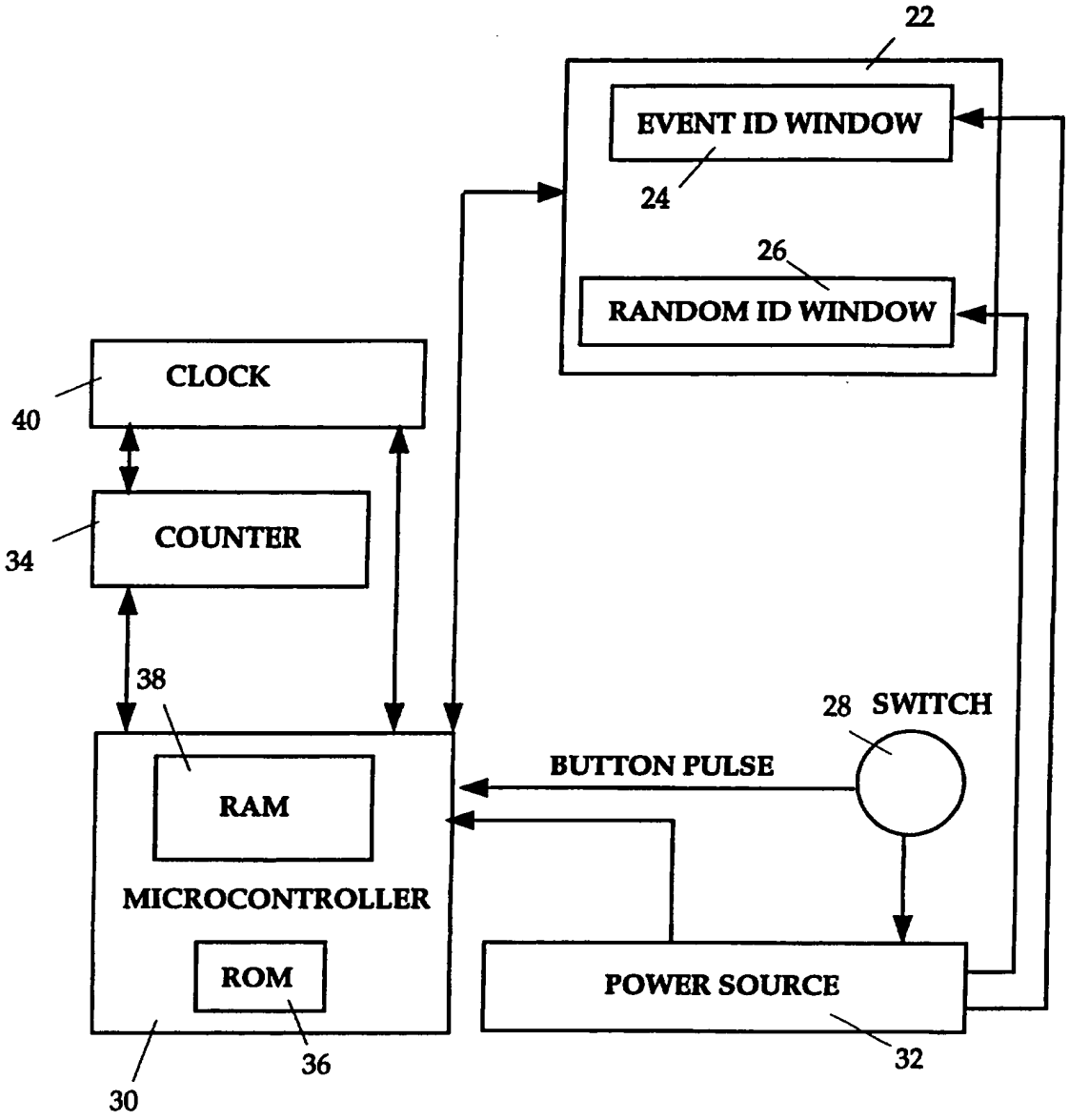

22
EVENT ID WINDOW
24
26
RANDOM ID WINDOW
CLOCK
40
COUNTER
34
38
28 SWITCH
BUTTON PULSE
RAM
MICROCONTROLLER
ROM
36
30
POWER SOURCE
32

FIG. 2B

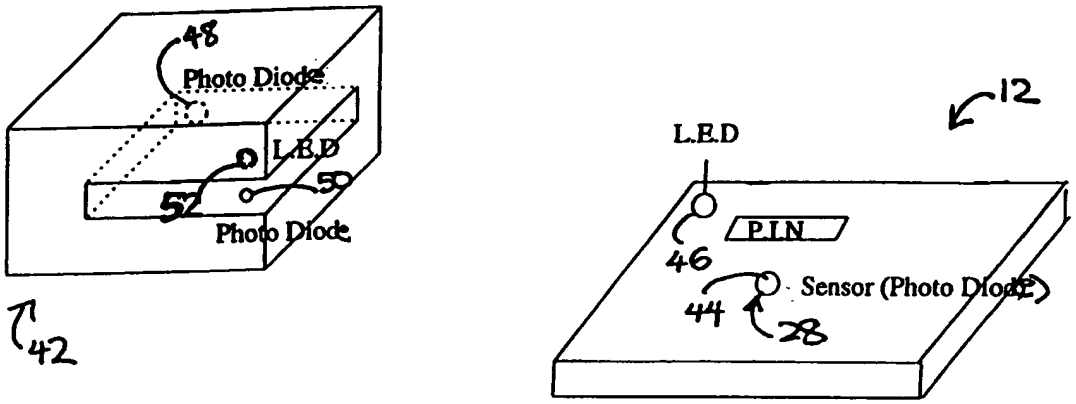

48
Photo Diode
L.E.D
52
50
Photo Diode
42
12
L.E.D
P.I.N
46
Sensor (Photo Diode)
44
28

FIG. 2C

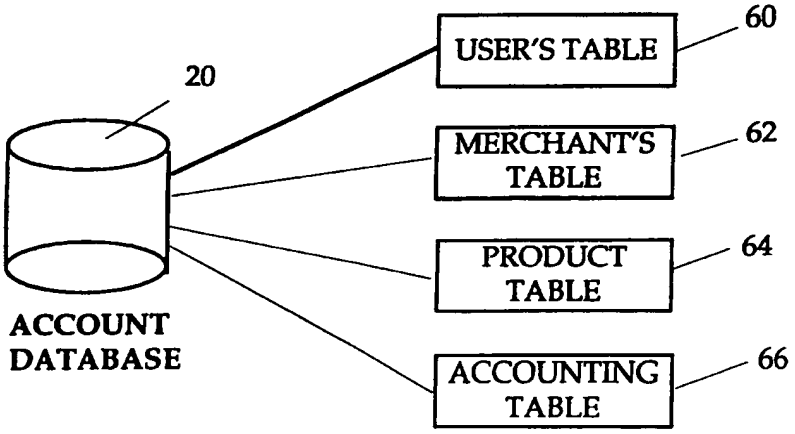

20
ACCOUNT DATABASE
USER'S TABLE
60
MERCHANT'S TABLE
62
PRODUCT TABLE
64
ACCOUNTING TABLE
66

FIG. 2D

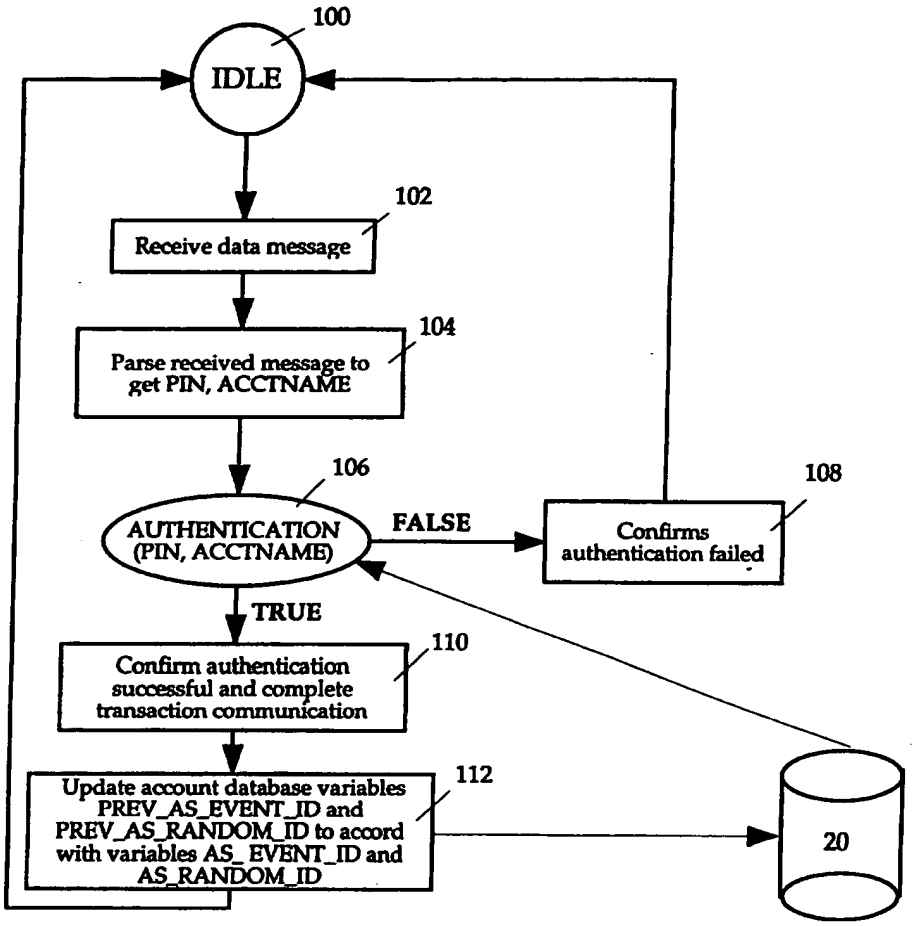

100
IDLE
102
Receive data message
104
Parse received message to get PIN, ACCTNAME
106
AUTHENTICATION (PIN, ACCTNAME)
FALSE
108
Confirms authentication failed
TRUE
110
Confirm authentication successful and complete transaction communication
112
Update account database variables PREV_AS_EVENT_ID and PREV_AS_RANDOM_ID to accord with variables AS_ EVENT_ID and AS_RANDOM_ID
20

FIG. 3A

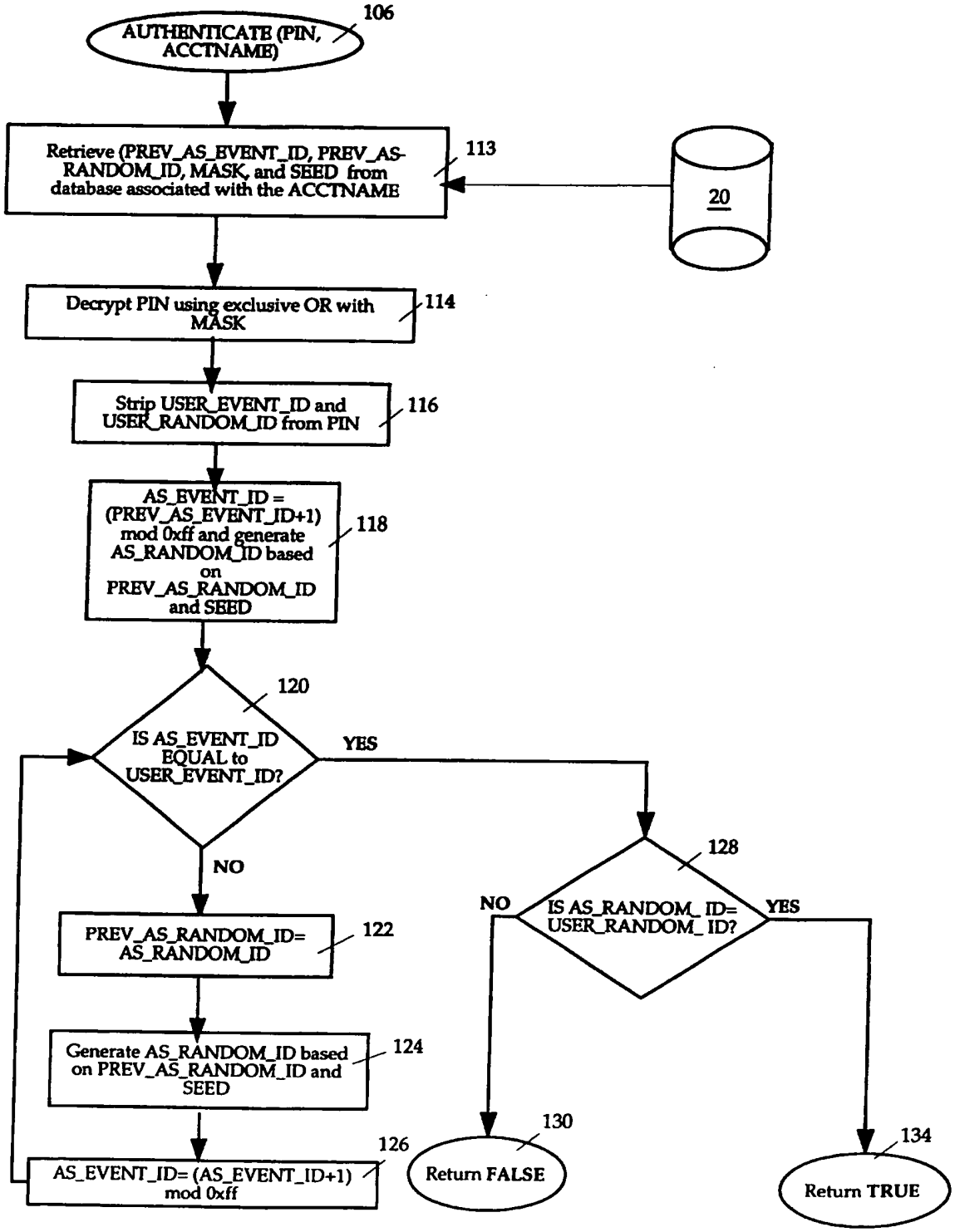

106
AUTHENTICATE (PIN, ACCTNAME)
Retrieve (PREV_AS_EVENT_ID, PREV_AS-RANDOM_ID, MASK, and SEED from database associated with the ACCTNAME
113
20
Decrypt PIN using exclusive OR with MASK
114
Strip USER_EVENT_ID and USER_RANDOM_ID from PIN
116
AS_EVENT_ID = (PREV_AS_EVENT_ID+1) mod 0xff and generate AS_RANDOM_ID based on PREV_AS_RANDOM_ID and SEED
118
120
IS AS_EVENT_ID EQUAL to USER_EVENT_ID?
YES
NO
128
NO
IS AS_RANDOM_ ID= USER_RANDOM_ ID?
YES
PREV_AS_RANDOM_ID= AS_RANDOM_ID
122
Generate AS_RANDOM_ID based on PREV_AS_RANDOM_ID and SEED
124
126
AS_EVENT_ID= (AS_EVENT_ID+1) mod 0xff
130
Return FALSE
134
Return TRUE

FIG. 3B

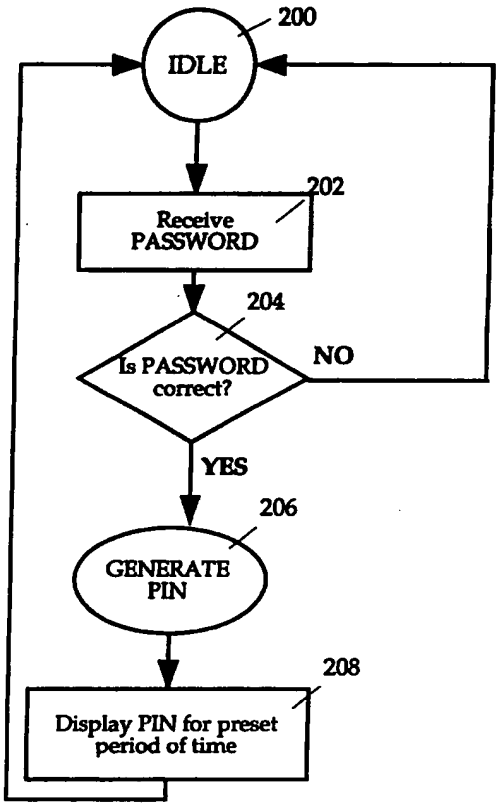

200
IDLE
202
Receive PASSWORD
204
Is PASSWORD correct?
NO
YES
206
GENERATE PIN
208
Display PIN for preset period of time

FIG. 4A

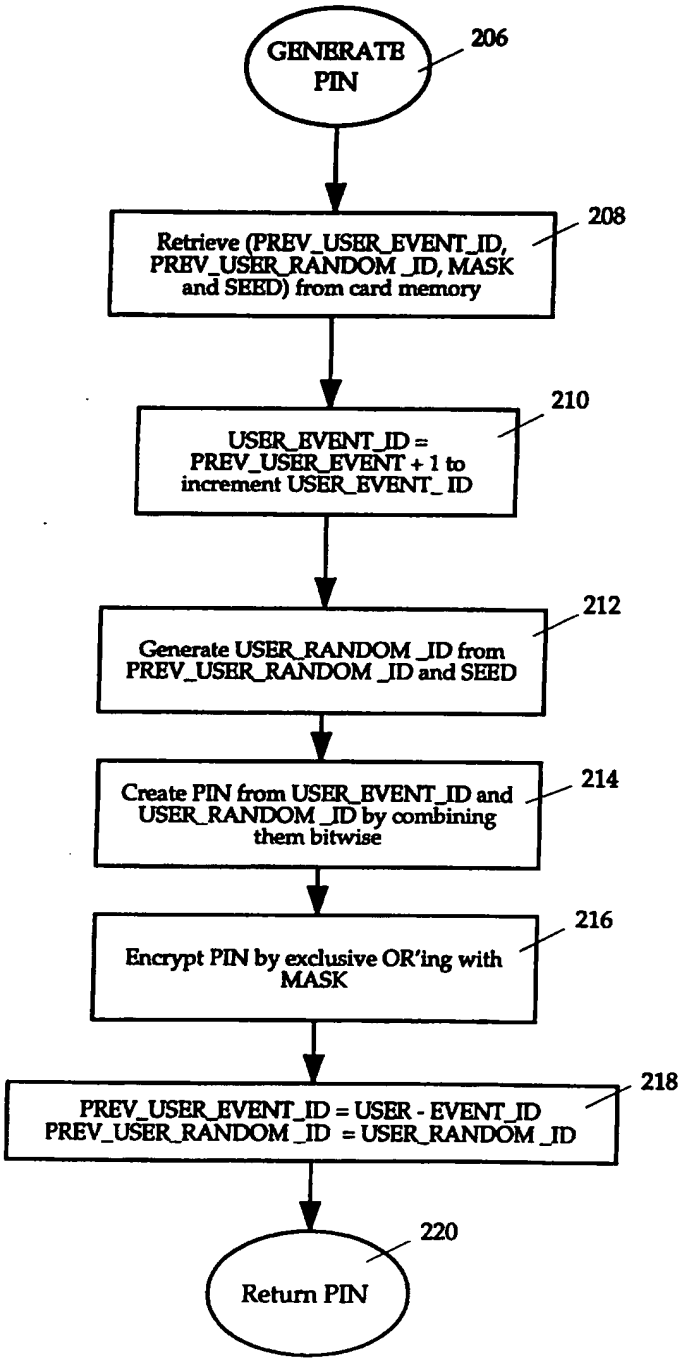

GENERATE PIN
206
Retrieve (PREV_USER_EVENT_ID, PREV_USER_RANDOM _ID, MASK and SEED) from card memory
208
USER_EVENT_ID = PREV_USER_EVENT + 1 to increment USER_EVENT_ ID
210
Generate USER_RANDOM _ID from PREV_USER_RANDOM _ID and SEED
212
Create PIN from USER_EVENT_ID and USER_RANDOM _ID by combining them bitwise
214
Encrypt PIN by exclusive OR'ing with MASK
216
PREV_USER_EVENT_ID = USER - EVENT_ID
PREV_USER_RANDOM _ID = USER_RANDOM _ID
218
Return PIN
220

FIG. 4B

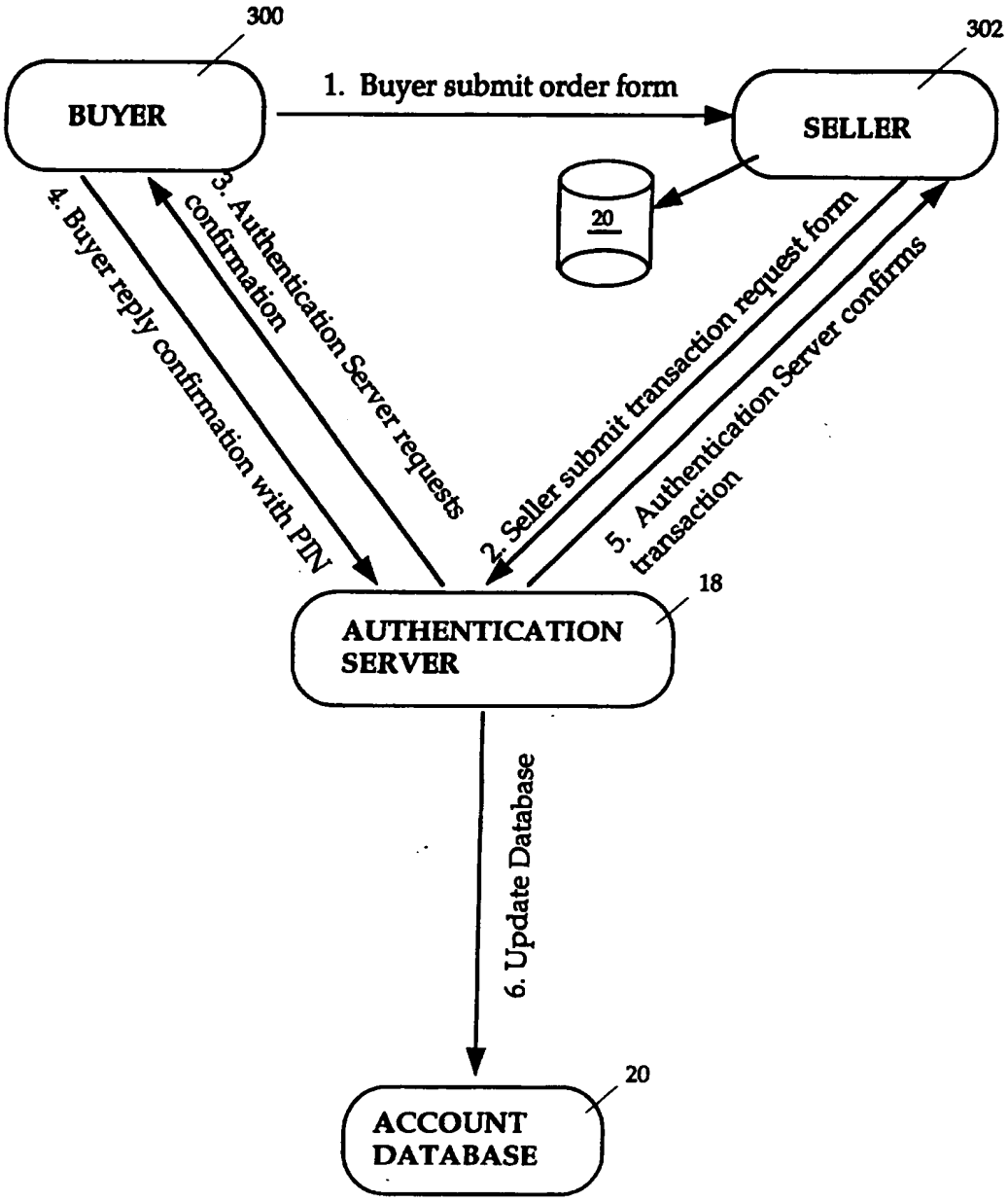

300
BUYER
1. Buyer submit order form
302
SELLER
20
3. Authentication Server requests confirmation
4. Buyer reply confirmation with PIN
2. Seller submit transaction request form
5. Authentication Server confirms transaction
18
AUTHENTICATION SERVER
6. Update Database
20
ACCOUNT DATABASE

FIG. 5

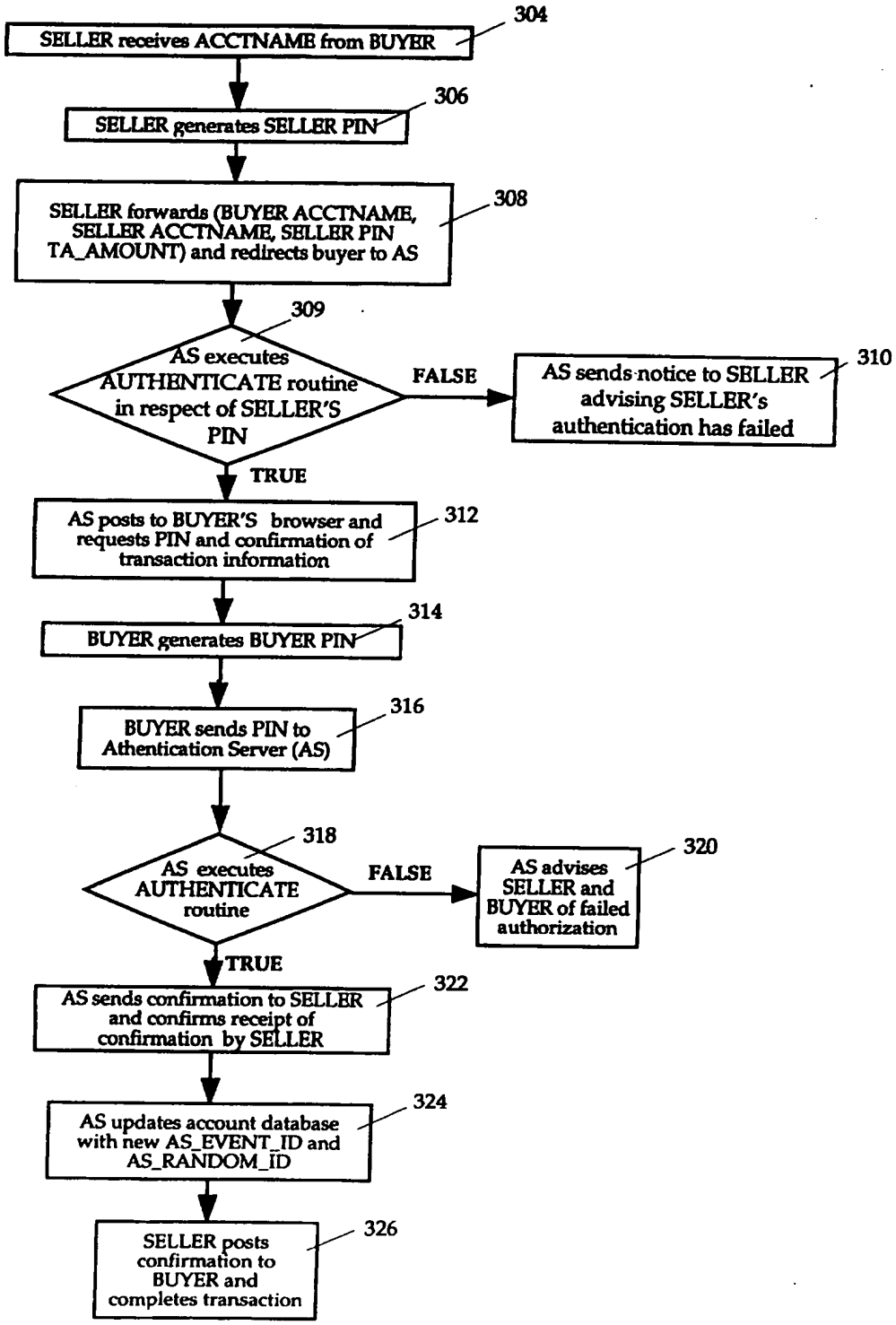

SELLER receives ACCTNAME from BUYER
304
SELLER generates SELLER PIN
306
SELLER forwards (BUYER ACCTNAME, SELLER ACCTNAME, SELLER PIN TA_AMOUNT) and redirects buyer to AS
308
309
AS executes AUTHENTICATE routine in respect of SELLER'S PIN
FALSE
AS sends notice to SELLER advising SELLER's authentication has failed
310
TRUE
AS posts to BUYER'S browser and requests PIN and confirmation of transaction information
312
BUYER generates BUYER PIN
314
BUYER sends PIN to Athentication Server (AS)
316
318
AS executes AUTHENTICATE routine
FALSE
AS advises SELLER and BUYER of failed authorization
320
TRUE
AS sends confirmation to SELLER and confirms receipt of confirmation by SELLER
322
AS updates account database with new AS_EVENT_ID and AS_RANDOM_ID
324
SELLER posts confirmation to BUYER and completes transaction
326

FIG. 6

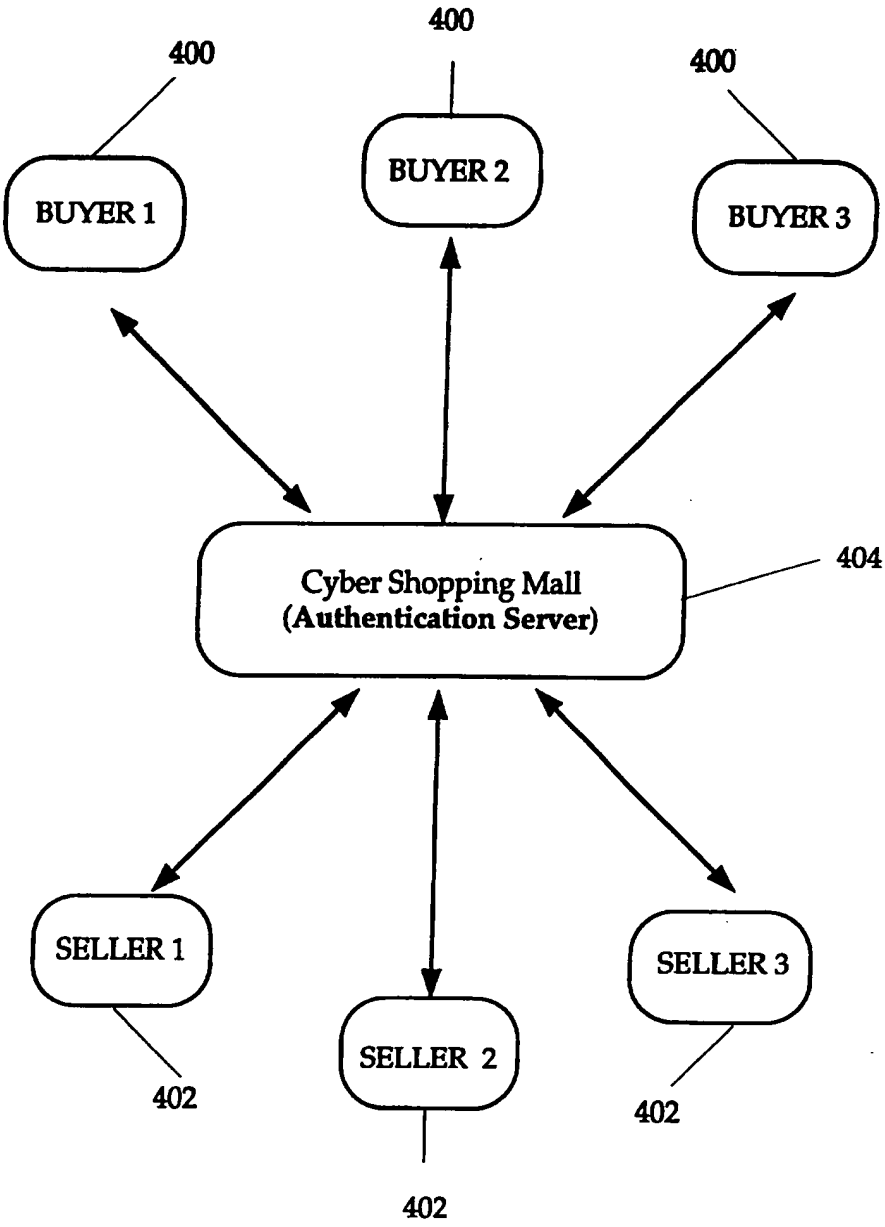

400
400
400
BUYER 1
BUYER 2
BUYER 3
Cyber Shopping Mall
(Authentication Server)
404
SELLER 1
SELLER 2
SELLER 3
402
402
402

FIG. 7

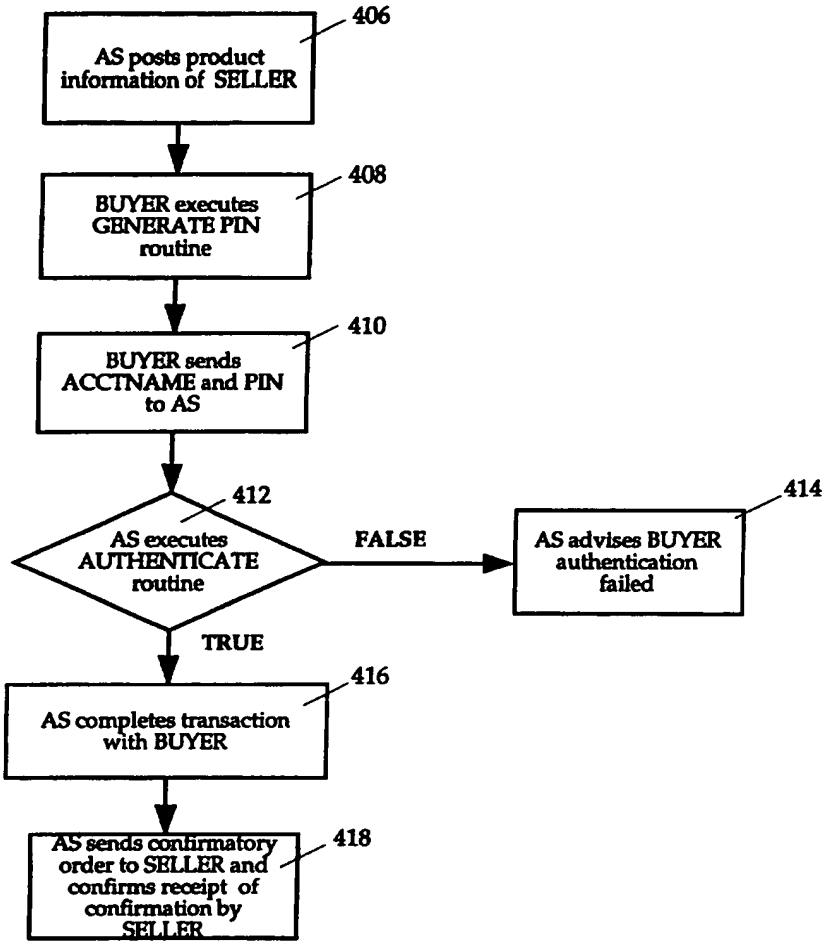

406
AS posts product information of SELLER
408
BUYER executes GENERATE PIN routine
410
BUYER sends ACCTNAME and PIN to AS
412
AS executes AUTHENTICATE routine
FALSE
414
AS advises BUYER authentication failed
TRUE
416
AS completes transaction with BUYER
418
AS sends confirmatory order to SELLER and confirms receipt of confirmation by SELLER

FIG. 8

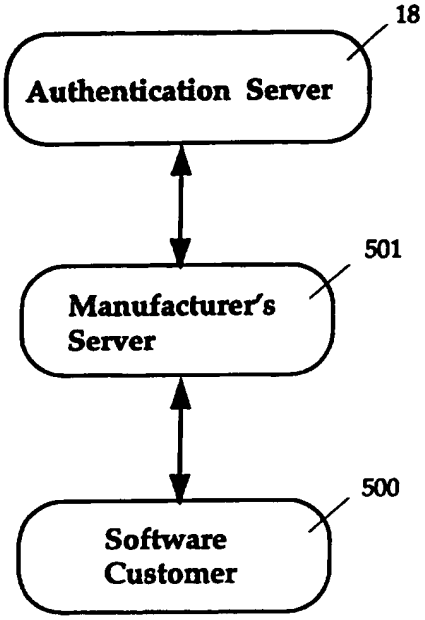

18
Authentication Server
501
Manufacturer's Server
500
Software Customer

FIG. 9

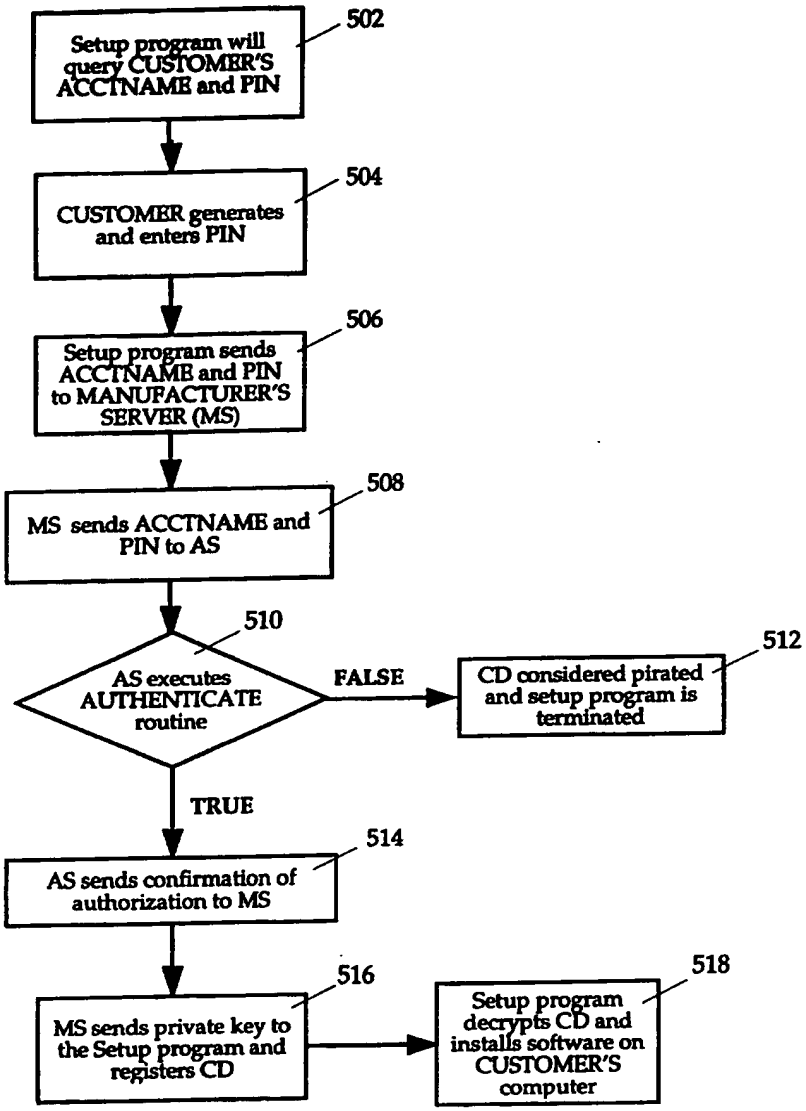

Setup program will query CUSTOMER'S ACCTNAME and PIN
502
CUSTOMER generates and enters PIN
504
Setup program sends ACCTNAME and PIN to MANUFACTURER'S SERVER (MS)
506
MS sends ACCTNAME and PIN to AS
508
AS executes AUTHENTICATE routine
510
FALSE
CD considered pirated and setup program is terminated
512
TRUE
AS sends confirmation of authorization to MS
514
MS sends private key to the Setup program and registers CD
516
Setup program decrypts CD and installs software on CUSTOMER'S computer
518

FIG. 10